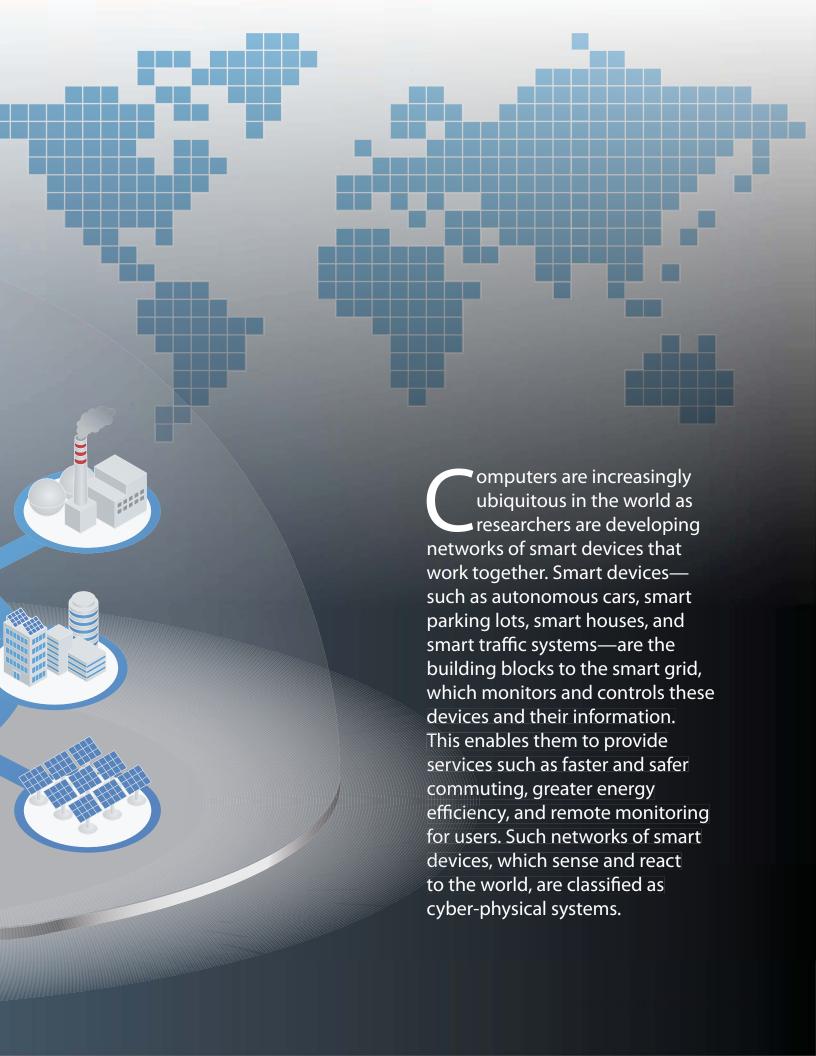
Resilient and secure cyber-physical systems

William Emfinger | Pranav Srinivas Kumar | Gabor Karsai





Cyber-physical systems

A cyber-physical system (CPS) is composed of embedded computing nodes communicating and collaborating to control aspects of its physical environment. The interaction between the computers and their environment, however, causes a range of complexities—in development, testing, verification, run-time, and management—which must be properly handled. These complexities compound the already difficult task of ensuring reliability and security of the CPS as a whole.

An example of such a CPS is a fractionated (i.e., divided) satellite cluster orbiting in formation. Each satellite provides communications, sensing, and computational resources to the rest of the cluster in fulfillment of the mission goals. It is difficult to develop the satellite system and the applications which control the system and run on it; this necessitates the use of system and software models for design-time analysis and verification of system performance and stability.

Such analysis includes the verification of operation deadline and timing properties, as well as network resource (e.g., buffer capacity and bandwidth) characteristics provided by the system and required by the applications running on the system. However, for such safety-, mission-, and security-critical systems, we must further ensure that the system is resilient to faults and anomalies. It must also be secure against attacks from compromised components within the system as well as from external sources.

CPS development process

To facilitate rapid and robust design and development of applications for our resilient CPS (RCPS) test bed (see figure 1), we developed an integrated tool suite for model-driven system and application development. Using this tool suite, we can describe—very precisely—the component-based software architecture for the applications which will run on the CPS in

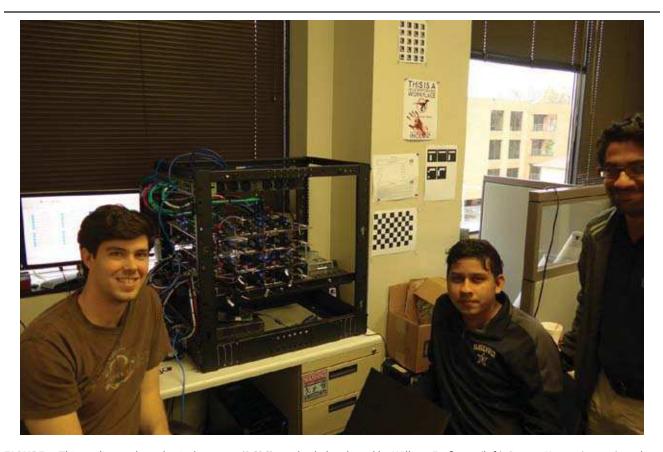


FIGURE 1. This resilient cyber-physical systems (RCPS) test bed, developed by William Emfinger (left), Pranav Kumar (center), and Amogh Kulkarni (right), performs security and resilience testing of CPS and their applications.

service of the mission's goals. These software models are developed in tandem with system models that incorporate the physics of the system. (For example, network characteristics such as bandwidth and latency between satellites vary periodically as functions of time according to the satellites' orbital parameters).

By composing these models together, we can analyze at design time the resource requirements and utilization of the applications on the system. Furthermore, the code generators we developed for the tool suite allow us to generate most of the application code (i.e., the infrastructural code which interfaces with the middleware) in a reliable manner. By relying on these code generators, we are able to focus on the core business-logic code, which provides the functionality we want from the applications on the cluster. These applications allow us to test the systems we are examining; for example, testing the detection and mitigation strategies for compromised or malicious software components based on the behavior of their network traffic.

Resilient CPS test bed

The RCPS test bed itself (see figure 2) is composed of the following components:

- ▶ 32 embedded Linux computers (BeagleBone Blacks) with ARMv7L architecture;
- OpenFlow-capable smart gigabit network switch, which allows the network characteristics of the system to be enforced on all network traffic;
- **physics simulation**, which allows the physical dynamics of the hardware to be simulated along with the sensor data and actuator control (for our satellite cluster system, we use Orbiter Space Flight Simulator);
- **standard gigabit network switch,** which allows fast communication (simulating the hardware bus) between the physics simulation and the nodes of the cluster; and
- **development machine**, which allows the modeling, development, deployment, and monitoring of the application code which runs the system.

By integrating the physics simulation and network emulation into the cluster (see figure 3), we are able to, in the case of the satellite cluster example, use the

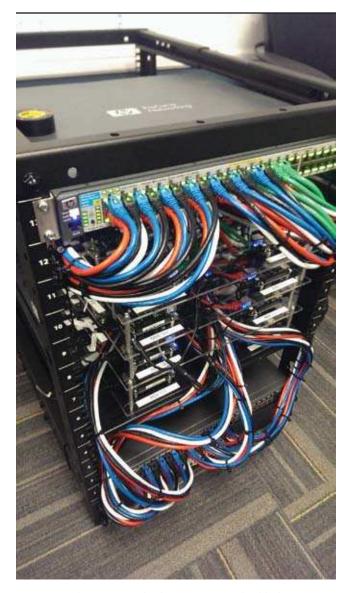


FIGURE 2. This RCPS test bed contains 32 embedded Linux computing boards. Each board (middle) is connected to both a smart network switch (top) which performs network emulation using OpenFlow and a regular network switch (bottom) that provides access to the physics simulation (Orbiter Space Flight Simulator).

physics simulation to determine the network characteristics between the nodes of the cluster. We can then enforce those characteristics (i.e., bandwidth, delay, and packet loss) on the cluster's network traffic through the smart switch. In this way, we can ensure that the applications running on the cluster see the same sensor and network behavior as they would in the real system. Because these types of mobile, networked CPSs are becoming more prevalent, the

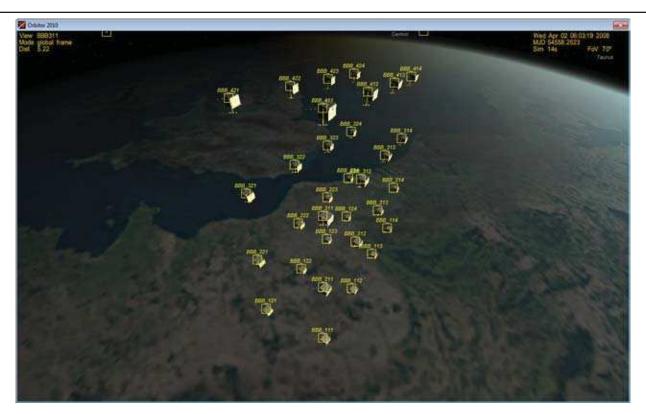


FIGURE 3. This Orbiter Space Flight Simulator simulation of the 32-node satellite cluster, which is controlled by the RCPS test bed, calculates the physics of the satellites and simulates their sensors and actuators.

network resources are becoming more crucial to the systems' functionality. Therefore, the emulation of the network is required to ensure high fidelity of application test results with respect to the run-time system.

We are currently developing test applications which use our research into network and timing analysis techniques to detect malicious software components at run-time and mitigate the effect of their attacks. Using these techniques, our infrastructure will provide a stable system, capable of detecting coordinated attacks from distributed software components (e.g., a denial-of-service (DDoS) attack from compromised or malicious software attempting to bring down a system node or an attack on specific sensors and actuators to make the system unstable).

Summary

We created the RCPS test bed as the foundational infrastructure for running experiments on CPSs and their software. A critical part of a CPS is the interaction with and feedback from the physical world, so

the integration of the physics simulation increases the fidelity of our cluster test results with respect to the system we are analyzing. The modeling, analysis, generation, deployment, and management tool suite we have developed drastically cuts down on the application development difficulty and time. This allows us to focus on the tests we want to run and the systems we want to analyze.

About the authors

William Emfinger has a PhD in electrical engineering from Vanderbilt University in Nashville, Tennessee. He received his BS in electrical engineering and biomedical engineering at Vanderbilt in 2011. During his undergraduate studies, Emfinger worked in rehabilitation engineering. His research interests combine cyber-physical/embedded systems engineering with high-performance and high-fidelity system simulation and rendering, with a focus on aerospace systems engineering. Besides these research interests, he enjoys theoretical physics and mathematics. He expects to complete his PhD studies in 2015.

Pranav Srinivas Kumar is a graduate research assistant at the Institute for Software Integrated Systems at Vanderbilt University. His research focuses on modeling, analysis, simulation and verification of component-based software applications executed on distributed real-time embedded systems. He received his BE in electronics and communication engineering from Anna University in Chennai, India in 2011.

Gabor Karsai is a professor of electrical and computer engineering at Vanderbilt University and senior research scientist at the Institute for Software Integrated Systems. He has over 20 years of experience in software engineering. Karsai conducts research in: the design and implementation of advanced software systems for real-time, intelligent control systems; programming tools for building visual programming environments; and the theory and practice of model-integrated computing. He received his BSc and MSc in electrical and computer engineering from the Technical University of Budapest in 1982 and 1984,

and he received his PhD in electrical and computer engineering from Vanderbilt University in 1988. Karsai has published over 160 papers, and he is the coauthor of four patents.

Acknowledgements

We would like to thank Dexter Watkins and Cameron Ridgewell for their assistance in the design and development of the RCPS test bed.

This work was supported by the Air Force Research Laboratory (AFRL) Science of Secure and Resilient Cyber-Physical Systems project under contract FA8750-14-2-0180 and by Vanderbilt University. We thank the sponsors for their generous support. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFRL or the US government.